St Mark's C of E Primary School: Medium Term Planning

Computing: Cycle B Spring Term

Predominant Area of Computing		
Computer Science	**Information Technology**	**Digital Literacy**

KS1

Unit Focus Introduction to Animation

NC Objectives
- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- create and debug simple programs
- use logical reasoning to predict the behaviour of simple programs
- use technology purposefully to create, organise, store, manipulate and retrieve digital content

	Learning Objective What is being learned rather than what is being done	**Resources** Any links to resources.	**End-point knowledge** What knowledge should children have gained by the end of the lesson? This will be the focus of retrieval activities and monitoring.
1	To choose a command for a given purpose **Teach Computing**	Scratch JR Optional- BeeBots	**Vocab: Bee-Bot, command, sprite, compare, programming, programming area** • I can find the commands to move a sprite • I can use commands to move a sprite • I can compare different programming tools
2	To show that a series of commands can be joined together **Teach Computing**	Worksheets Scratch JR	**Vocab: Block, joining, command, Start block, run, program, programming area, background, delete, reset, algorithm, predict** • I can use more than one block by joining them together • I can use a **Start** block in a program • I can run my program]

3	To identify the effect of changing a value **Teach Computing**	Worksheets Scratch JR	**Vocab: Effect, change, value, block** • I can find blocks that have numbers • I can change the value • I can say what happens when I change a value
4	To explain that each sprite has its own instructions **Teach Computing**	Worksheets Scratch JR	**Vocab: Instructions, sprite, delete, program, algorithm** • I can show that a project can include more than one sprite • I can delete a sprite • I can add blocks to each of my sprites
5	To design the parts of a project **Teach Computing**	Worksheets Scratch JR	**Vocab: Sprite, background, appropriate, algorithm** • I can choose appropriate artwork for my project • I can decide how each sprite will move • I can create an algorithm for each sprite
6	To use my algorithm to create a program **Teach Computing**	Worksheets Scratch JR	**Vocab: Sprite, design, programming blocks, algorithm, programs** • I can use sprites that match my design • I can add programming blocks based on my algorithm • I can test the programs I have created

LKS2

Unit Focus Coding

NC Objectives

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

	Learning Objective What is being learned rather than what is being done	**Resources** Links to resources.	**End-point knowledge** What knowledge should children have gained by the end of the lesson – this will be the focus of retrieval
1	Using Flowcharts **3.1 PM**	Chromebooks/ Laptops	**Vocab: algorithm, object, background, attributes, action, implement, flowchart, event scene.** • Children can read and explain a **flowchart**. • Children can use a **flowchart** to create a computer program. • Children can create a computer program that includes **click events** and **timers**.
2	**Using Timers** **3.1 PM**	Chromebooks/ Laptops	**Vocab: sequence, nested, timer, interval** • Children can create a program that uses a timer-after command • Children can create a program that uses a timer-every command • Children understand there can be different ways to solve a problem
3	Using Repeat **3.1 PM**	Chromebooks/ Laptops	**Vocab: repeat, degrees, right-angle, turtle object** • Children understand how the turtle object moves. • Children can use the repeat command with an object. • Children can create a computer program that includes use of the repeat command.
4	Code, Test and Debug **3.1 PM**	Chromebooks/ Laptops	**Vocab: debug, code, test, nesting** • Children can create computer programs using prior knowledge. • Children can run, test and debug their programs. • Children can consider nesting when debugging their programs.
5/ 6.	Design and Make an Interactive Scene **3.1 PM**	Chromebooks/ Laptops	**Vocab: object, attributes, click-events, alert, deb, debugging** • Children can use the attributes (properties) table to set the attributes of objects. • Children can plan their scene and code before they create their program. • Children can confidently make several different things happen in a program.

UKS2

Unit Focus Digital Science – Sensing Movement

NC Objectives

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

	Learning Objective What is being learned rather than what is being done	**Resources** Links to resources.	**End-point knowledge** What knowledge should children have gained by the end of the lesson – this will be the focus of retrieval
1	To create a program to run on a controllable device **Teach Computing**	Laptops Micro:bit	**Vocabulary: algorithm, code, Micro:bit.** • I can apply my knowledge of programming to a new environment • I can test my program on an emulator • I can transfer my program to a controllable device
2	To explain that selection can control the flow of a program **Teach Computing**	Laptops Micro:bit Make code – online code editor	**Vocabulary: variable** • I can identify examples of conditions in the real world • I can use a variable in an if, then, else statement to select the flow of a program • I can determine the flow of a program using selection
3	To update a variable with a user input **Teach Computing**	Laptops Micro:bit	**Vocabulary: input, value** • I can use a condition to change a variable • I can experiment with different physical inputs • I can explain that checking a variable doesn't change its value
4	To use a conditional statement to compare a variable to a value **Teach Computing**	Laptops Micro:bit	**Vocabulary: operand, modify** • I can use an operand (e.g. <>=) in an if, then statement • I can explain the importance of the order of conditions in else, if statements • I can modify a program to achieve a different outcome

5.	To design a project that uses inputs and outputs on a controllable device **Teach Computing**	Laptops Micro:bit	**Vocabulary: variable, algorithm** ● I can decide what variables to include in a project ● I can design the algorithm for my project ● I can design the program flow for my project
6.	To develop a program to use inputs and outputs on a controllable device **Teach Computing**	Laptops Micro:bit	**Vocabulary: input, output** ● I can create a program based on my design ● I can test my program against my design ● I can use a range of approaches to find and fix bugs